Aplicaciones documentales de la recuperación de información

aplicaciones prácticas para su mejor explotación documental

Manuel Blázquez Ochando

Monografías electrónicas

mblazquez.es

025.4.036:02 **BLA** **apl**	BLÁZQUEZ OCHANDO, Manuel Aplicaciones documentales de la recuperación de información: aplicaciones para su mejor explotación documental / Manuel Blázquez Ochando.– Madrid: mblazquez.es, 2012. 71p. ; 21cm.– (Libros y manuales de la Documentación; 1) ISBN 978-84-695-6372-4 1. Biblioteconomía y Documentación 2. Recuperación de Información 3. Tecnologías de la Documentación I. Título II. Series

UNIVERSIDAD COMPLUTENSE DE MADRID
Facultad de Ciencias de la Documentación

1ªed. noviembre 2012, Madrid

Publicado por mblazquez.es

ISBN 978-84-695-6372-4

Índice

1. Introducción

La continua proliferación y crecimiento de la información publicada en la red, hace necesario un conocimiento más profundo de las técnicas, herramientas y aplicaciones en recuperación de información. En este sentido los sistemas de gestión de contenidos también denominados CMS (Como Joomla o Drupal) han contribuido a facilitar la organización de la información y al mismo tiempo multiplicar el número de vías y medios de acceso a la misma. En este marco de trabajo también se circunscriben los sistemas de redifusión o sindicación de contenidos, así como las herramientas y modelos de recuperación.

Se consideran aplicaciones documentales en su sentido más amplio, todas aquellas herramientas cognitivas de tipo clasificatorio, librario o informático que facilitan y ayudan al documentalista en su actividad profesional. En el contexto en el que se abordará la asignatura, en el de la recuperación de información, se consideran aplicaciones documentales a los sistemas de redifusión y recuperación de información bibliográfica sindicada (OrangeUp), sistemas de recuperación basados en técnicas de agrupación o clustering (Carrot2), la metodología de consulta en bases de datos SQL, sistemas de recuperación con expansión de consulta, los sistemas de indexación y análisis de contenidos a gran escala (OmniFind), así como a los motores de recuperación de alto rendimiento como (Apache Lucene).

2. Recuperación de información en bases de datos

Qué es una base de conocimiento

Es cualquier colección o fondo documental que constituye el corpus de un sistema de recuperación de información. Habitualmente esta base de conocimiento se organiza y estructura en bases de datos para su mejor gestión, tratamiento y recuperación. Esto significa que base de conocimiento puede ser desde un compendio de datos, cifras y cadenas de texto inconexas, hasta documentos, referencias bibliográficas y compendios informativos y semánticos con plena significación.

Qué es una base de datos

La base de datos es el sistema que posibilita la organización y estructuración de los contenidos o bases de conocimiento en tablas y éstas a su vez en campos, de tal forma que cada campo represente una característica o rasgo descriptivo de la información o contenido registrado en la base de datos y cada tabla represente el dominio general que se está almacenando. Por ejemplo una tabla de usuarios contendrá campos lógicos que definan, describan e identifiquen a cada usuario. Por ejemplo el nombre, apellidos, DNI, dirección, correo, sitio web, teléfono, código postal, etc. Dentro de las distintas tablas de una base de datos es posible encontrar relaciones evidentes, ampliando la magnitud de la información. Es el caso de las bases de datos relacionales. Por ejemplo la tabla usuarios puede estar relacionada con la tabla préstamos en la que se relacionan los documentos y materiales librarios que se les presta. Esta operación en todo caso requiere de un campo clave de relación, que puede ser el identificador del usuario, sobre el que se registra los datos del préstamo y el identificador del libro que se le está prestando. Este mecanismo tan sencillo hace posible que distintas tablas queden vinculadas y puedan ser contrastadas y filtradas. Pero una base de datos comporta muchos más aspectos, el tipo de campos, sus características especiales para almacenar determinados contenidos, por ejemplo datos binarios, imágenes, textos de gran extensión, numeración en coma flotante, etc. Todos estos componentes hacen que cualquier base de conocimiento pueda ser recogida sea cual sea su naturaleza y características.

Qué es un gestor de bases de datos

El manejo de las bases de datos habitualmente se lleva a cabo mediante comandos bien definidos en terminales especializados, shell (Linux), cmd (Windows). Estos comandos en la mayoría de los casos responden al lenguaje de consulta normalizado SQL (Structured Query Language) con el que la base de datos entiende qué debe hacer. Mediante este lenguaje es posible dar órdenes al sistema para que inserte un registro, lo borre, lo edite y por supuesto recupere un determinado dato, información o documento. Dado que este método de comunicación requiere un tecleado continuo para interactuar con el sistema, se han desarrollado programas informáticos que llevan a cabo dicha función de manera automática, facilitando al administrador un interfaz gráfico para la edición, tratamiento y recuperación de la información. Estos programas que permiten trabajar con las bases de datos, las tablas, los campos y los datos almacenados en ellas se denominan gestores de bases de datos. Uno de los más conocidos y utilizados en todo el mundo es PhpMyAdmin. Diseñado para trabajar principalmente con bases de datos que emplean el lenguaje SQL.

Qué es MySQL

MySQL es la principal base de datos que alambica la web. Utiliza el lenguaje de consulta SQL y es utilizada conjuntamente con el lenguaje de programación PHP para crear las principales aplicaciones de la red. Normalmente actúa como un componente más que se instala en el paradigma de desarrollo web WAMP (SO. Windows, Servidor http Apache, BD MySQL, intérprete PHP), LAMP (SO. Linux, Servidor http Apache, BD MySQL, intérprete PHP) o MAMP (SO. Mac, Servidor http Apache, BD MySQL, intérprete PHP). Dicho de otra forma es la base de datos que sirve para almacenar la mayoría de los datos y transacciones comunicativas que se producen en internet. Es muy notable su utilización en Sitios Web, CMS (Content Management System) o gestores de contenidos, Sistemas de Gestión Integral de Bibliotecas, Archivos, Museos y un largo etcétera de herramientas y aplicaciones de software libre. Entre sus características más importantes destaca su capacidad para ejecutar múltiples consultas en distintos hilos de ejecución por segundo, gran capacidad de almacenamiento y motor de almacenamiento que efectúa un proceso de indexación automático de los contenidos y en todo caso de la base de conocimiento con que se alimente. Estas últimas características permiten hablar de recuperación de información y no de recuperación de

datos, ya que efectúa por si solo los procesos de tratamiento previos a la recuperación de información.

- Eliminación de palabras vacías
- Indexación de contenidos
- Creación de fichero inverso
- Análisis de frecuencias de cada término
- Recuperación booleana
- Recuperación a texto completo
- Recuperación con lenguaje natural
- Recuperación con expansión de consulta

Cómo funciona

Para trabajar con MySQL se necesita previamente instalarlo. Esta tarea se lleva a cabo con la instalación de un servidor http y sus componentes habituales, utilizando uno de los paradigmas de desarrollo web mencionados anteriormente. Por ejemplo mediante la instalación de una distribución WAMP (AppServ), LAMP (Xampp) o MAMP (mamp). Una vez instalado se requieren unos datos de conexión fundamentales para empezar a operar, estos son:

- Nombre del servidor. Esto es la dirección IP o el nombre de dominio o la máquina en el que está instalado el servicio HTTP, la base de datos MySQL y el intérprete PHP. Por ejemplo si se trata de una instalación local que corresponde a nuestro equipo u ordenador, la dirección IP del servidor siempre será 127.0.0.1 y el nombre de servidor localhost, normalmente.

- Nombre de la base de datos. Para que MySQL sepa con que base de datos se va a trabajar, es necesario indicar o seleccionar en todo caso su nombre. Se debe pensar que MySQL puede generar tantas bases de datos como requiera el usuario, lo que implica que en todo momento es necesario distinguir sobre cuál se desea operar.

- Usuario y contraseña MySQL. Es el nombre del usuario y su contraseña de acceso a la base de datos MySQL. Éste puede ser un simple usuario con unos

privilegios limitados o el administrador del sistema. En una instalación local de prueba, tanto el usuario como la contraseña siempre será root (Convenido por defecto).

```
<?php
$con = mysql_connect('localhost', 'root', 'root') or die ('error: no se pudo conectar a mysql');
$database = 'openbiblio';
?>
```

Tabla 1. Ejemplo de sintaxis de conexión en PHP

Especificados los datos de conexión e introducidos en el sistema MySQL, se hace posible su consulta y operación mediante sentencias SQL o bien empleado el interfaz gráfico de un gestor de bases de datos como el mencionado anteriormente.

3. Principios de SQL y sintaxis básica

Por lo tanto SQL es un lenguaje de consulta estándar diseñado para operar en bases de datos. Como se ha explicado actúa en MySQL, pero también puede operar en otras bases de datos como Oracle, DB2, SQL server, PostgreSQL, etc. Qué operaciones permite llevar a cabo:

- Ejecutar consultas y recuperar datos
- Efectuar procesos de recuperación de información
- Insertar, actualizar y eliminar registros
- Crear nuevas bases de datos, tablas y campos
- Establecer permisos de administración para los usuarios
- Crear distintas vistas de una base de datos

Aprendiendo la sintaxis básica

En este curso nos centraremos en los métodos de recuperación por medio de SQL. Pero para ello es necesario aprender una sintaxis básica con la que se explican los principios de consulta SQL. En la siguiente tabla2 se observan una serie de palabras reservadas o cláusulas (coloreadas en rojo) que corresponden al selector de campos (SELECT), al selector de tablas (FROM) y a la cláusula condicional (WHERE).

```
SELECT campos FROM tabla WHERE condición
```

Tabla 2. Sintáxis de consulta básica

Siempre que se desea obtener datos o resultados de una consulta SQL se requiere un selector de los campos que son objetivo de la búsqueda, seleccionar la tabla en la que se desea buscar la información y establecer las condiciones oportunas que deben cumplir los resultados. Véase el siguiente ejemplo de la tabla3.

```
SELECT isbn FROM catalogo WHERE autor LIKE '%bryson%'
```

Tabla 3. Ejemplo de consulta de todos los isbn del catálogo de libros cuyo autor sea bryson

Este ejemplo se puede traducir de la siguiente forma: Selecciona el campo *isbn* de todos los registros de la tabla *catálogo* que cumplan la condición de que dentro del campo *autor* se contenga el término *bryson*. Obsérvese que el término de consulta está rodeado de porcentajes. Esos caracteres también denominados truncamientos actúan

sobre la consulta para indicar que la cadena bryson puede tener cadenas de texto que le precedan y que le sigan, como por ejemplo *bill* ***bryson*** *natural de*.

Crear una base de datos

Para crear una base de datos solo es necesario recordar la fórmula (CREATE DATABASE + nombre de la base de datos), véase tabla4. El nombre de la base de datos es recomendable que se escriba siempre en minúsculas, sin caracteres extraños, símbolos o acentos. Tampoco debería preceder al nombre ningún número. Por otro lado los espacios en el nombre deben ser sustituidos por guiones bajos (_) o guiones medios (-). Finalmente es recomendable el uso de nombres sencillos que puedan ser fácilmente recordados.

```
CREATE DATABASE biblioteca
```

Tabla 4. Crear una base de datos denominada "biblioteca"

Crear una tabla con campos

La creación de una tabla en mysql implica también el diseño de su estructura de campos y con ello definir sus características. Resulta vital la forma en que se diseñan las tablas para así poder efectuar una mejor recuperación y utilizar funciones específicamente diseñadas para tal propósito, es el caso de las consultas de tipo FULLTEXT. Cuando se diseña la tabla, se deben establecer el tipo de campos que la componen en función al tipo de información que albergarán y a su extensión, por otro lado su set de codificación o set de caracteres que se utilizará, véase tabla5.

```
CREATE TABLE users (
id          INT NOT NULL AUTO_INCREMENT, PRIMARY KEY(id),
name        VARCHAR(255) CHARACTER SET utf8 COLLATE utf8_general_ci,
surname     VARCHAR(255) CHARACTER SET utf8 COLLATE utf8_general_ci,
allvisits   LONGTEXT CHARACTER SET utf8 COLLATE utf8_general_ci,
lastvisit   VARCHAR(255) CHARACTER SET utf8 COLLATE utf8_general_ci,
lastsession VARCHAR(255) CHARACTER SET utf8 COLLATE utf8_general_ci,
online      VARCHAR(50) CHARACTER SET utf8 COLLATE utf8_general_ci,
level       VARCHAR(255) CHARACTER SET utf8 COLLATE utf8_general_ci,
username    VARCHAR(255) CHARACTER SET utf8 COLLATE utf8_general_ci,
password    VARCHAR(255) CHARACTER SET utf8 COLLATE utf8_general_ci,
trash       VARCHAR(500) CHARACTER SET utf8 COLLATE utf8_general_ci,
snumber     VARCHAR(2) CHARACTER SET utf8 COLLATE utf8_general_ci,
email       VARCHAR(255) CHARACTER SET utf8 COLLATE utf8_general_ci
) CHARACTER SET utf8 COLLATE utf8_general_ci
```

Tabla 5. Crear una tabla denominada "users"

Insertar un nuevo registro en la tabla

Crear un nuevo registro en la tabla anterior se lleva a cabo con la sentencia de la tabla6. Obsérvese la sintaxis (INSERT INTO + nombre de tabla afectada + SET + nombre del campo = 'datos', nombre del campo = 'datos', nombre del campo = 'datos'...) Esta forma de insertar datos permite al operador de MySQL tener un mayor control sobre la información que inserta en el registro, pues no esta obligado a introducir todos los datos de todos los campos si no lo desea. Es posible introducir sólo la información de los campos que se reseñen.

```
INSERT INTO users SET name='nombre', surname='apellidos', allvisits='registro de todas las visitas', lastvisits='última visita', lastsession='última sesión',online='estado', level='nivel de acceso', username='nombre de usuario',password='contraseña', trash='código de encriptación', snumber='código de seguridad', email='correo electrónico'
```

Tabla 6. Ejemplo de inserción de un registro completo en la tabla

Modificar y actualizar un registro de la tabla

Se utiliza la sintaxis (UPDATE + nombre de tabla afectada + SET + nombre del campo = 'nuevo dato', nombre del campo = 'nuevo dato', nombre del campo = 'nuevo dato' + WHERE + condición) Al igual que en el caso anterior de la inserción, no se está obligado a repetir todos los campos de la estructura que conforma la tabla de MySQL. Es suficiente reseñando sólo aquellos campos del registro en el que se van a suceder los cambios con nuevos datos. Finalmente se requiere la condición de la consulta de actualización, es decir, qué registro es el que se desea actualizar. En el caso de la tabla7, es sobre un registro en concreto, por lo que es necesario expresar que el identificador del usuario sea igual al que establezcamos.

```
UPDATE users SET name = 'nuevo nombre', surname = 'nuevos apellidos' WHERE id = 'identificador del registro'
```

Tabla 7. Ejemplo de modificación y actualización de un registro de una tabla

Eliminar un registro de la tabla

Para eliminar un registro de una tabla se emplea la sintaxis (DELETE FROM + nombre de tabla afectada + WHERE + condición), véase tabla8. Al igual que en el caso de la modificación y actualización de una tabla es preciso determinar la condición bajo la que

se eliminará los registros o registro concreto. Una vez más señalando el número de identificación es suficiente para indicar a MySQL cómo proceder.

```
DELETE FROM items WHERE id = 'identificador del registro'
```

Tabla 8. Borrar un registro de una tabla

4. Operaciones de consulta SQL esenciales

Hasta el momento se han advertido los mecanismos más sencillos de interactuación en lenguaje SQL. En este apartado se comenzará a explorar las posibilidades de consulta esenciales de SQL, concretamente el uso del operador LIKE, REGEXP (de comparación de cadenas) y los operadores AND, OR, XOR y NOT (booleanos).

Consulta con operador LIKE

El operador LIKE al igual que REGEXP, son operadores cuya misión es la comparación de cadenas o patrones dados en la consulta. La principal diferencia es que LIKE efectúa un reconocimiento de la cadena de consulta de forma absoluta a no ser que se especifiquen los caracteres precedentes y antecedentes con:

- (%) Establece coincidencia con cualquier extensión y tipo de caracter delante o detrás de la cadena de consulta, según se ubique el porcentaje.

- (_) Establece coincidencia con 1 caracter de cualquier tipo delante o detrás de la cadena de consulta, según se ubique el guión bajo.

```
SELECT * FROM catalogo WHERE titulo LIKE '%cupe%'
```

Tabla 9. Buscar cualquier registro de la tabla catálogo cuyo título contenga la cadena cupe. Esta consulta obtendría como resultado registros con la palabra recuperación, irrecuperable, ocupe, desocupen, etc.

```
SELECT * FROM catalogo WHERE isbn LIKE '978-84-____-___-5'
```

Tabla 10. Buscar cualquier registro de la tabla catálogo cuyo isbn contenga cualquier caracter entre 978-84- y -5. Por ejemplo se obtendría como resultado 978-84-1234-123-5.

Consulta con operador AND (&&)

Recupera registros siempre y cuando se cumplan todas las condiciones establecidas. Si alguna no se cumple, no se incluye como resultando, saltando a los siguientes registros de la tabla. Por ejemplo la consulta de la tabla11, establece que los registros deberán cumplir la condición de contener en su título, subtítulo, resumen, descripción y autor la palabra texto. Obsérvese la sintaxis empleada (SELECT + campos + FROM + tabla afectada + WHERE + condición1(campo + LIKE + 'texto de consulta') AND

condición2(campo + LIKE + 'texto de consulta') AND condición3(campo + LIKE + 'texto de consulta'))

```
SELECT * FROM catalogo WHERE
  titulo LIKE '%texto%' AND    subtitulo LIKE '%texto%' AND
  resumen LIKE '%texto%' AND
  descripcion LIKE '%texto%' AND
  autor LIKE '%texto%'
  LIMIT 0,30
```

Tabla 11. Consulta utilizando el operador AND

Consulta con operador OR (||)

Recupera registros siempre que al menos una de las condiciones establecidas se verifique. Por ejemplo, en la tabla12 se establece que los registros deberán cumplir que o bien el título coincida con el patrón, o bien sea el subtítulo, el resumen, descripción o autor. No devolverá ningún registro si no se cumple al menos una de las condiciones.

```
SELECT * FROM catalogo WHERE
  titulo LIKE '%texto%' OR    subtitulo LIKE '%texto%' OR
  resumen LIKE '%texto%' OR
  descripcion LIKE '%texto%' OR
  autor LIKE '%texto%'
  LIMIT 0,30
```

Tabla 12. Consulta utilizando el operador OR

Consulta con operador XOR

Variante absoluta del operador OR, recupera registros que cumplan una condición u otra pero nunca recuperará registros en que ambas condiciones se cumplan a la vez. Por ejemplo la consulta de la tabla13 recuperará cualquier libro del catálogo cuya temática sea arquitectura o bibliotecas, pero nunca arquitectura de bibliotecas.

```
SELECT * FROM catalogo WHERE
  tematica LIKE '%arquitectura%' XOR
  tematica LIKE '%bibliotecas%'
  LIMIT 0,30
```

Tabla 13. Consulta utilizando el operador XOR

Consulta con operador NOT (!)

Se utiliza como operador de precedencia con los operadores de comparación de cadenas de tipo LIKE, permite establecer una negación en las consultas. Por ejemplo en la

tabla14, se recuperará todos los registros del catálogo cuyo autor sea Cervantes y cuyos títulos no contengan la palabra Quijote. El resultado probable sería La Gitanilla, Rinconete y Cortadillo, El Licenciado Vidriera, etc., menos El Quijote.

```
SELECT * FROM catalogo WHERE
  autor LIKE '%cervantes%' AND
  title NOT LIKE '%quijote%'
  LIMIT 0,30
```

Tabla 14. Consulta utilizando el operador NOT

Consulta con operador REGEXP

REGEXP es un operador especializado en la comparación de cadenas de texto mediante expresiones regulares (REGular EXPressions). Las expresiones regulares se utilizan para afinar de forma mucho más precisa la consulta de datos o cadenas de texto. Estas se componen a base de caracteres especiales, los más comunes son:

- (.) Cada punto corresponde a un caracter individual, puede ser un número, letra o cualquier otro caracter.
- (^) Indica que el patrón comienza por la instrucción que siga a ^.
- ($) Indica que el patrón finaliza con la cadena o instrucción que preceda a $.
- (...) Tantos puntos se indiquen, tantos caracteres tendrá la cadena de texto objetivo.
- (^[Az]{5}) Indica que el patrón a buscar comienza por cualquier letra, con una extensión de 5 caracteres.
- (^.*[0-9]{2}$) Establece que la cadena comienza por cualquier caracter repetido n veces y que finaliza con un número de 2 cifras.

Por ejemplo en la tabla15 se consultan todos los autores cuyo apellido comience por Bal. El resultado podrá ser muy diverso, Balzac, Balz, Balza, etc.

```
SELECT * FROM catalogo WHERE
autor REGEXP '^Bal'
LIMIT 0,30
```

Tabla 15. Consulta utilizando el operador REGEXP

Referencias

- FRIEDL, J. 2006. Mastering Regular Expressions: Understand Your Data and Be More Productive. Disponible en: http://www.minek.com/files/Mastering%20Regular%20Expressions.pdf

- SKINNER, G. 2011. RegExr. Disponible en: http://gskinner.com/RegExr/

5. Recuperación avanzada con SQL

La base de datos MySQL dispone de una serie de funciones de recuperación de información para aquellos campos definidos como de tipo FULLTEXT. Esto significa que MySQL es capaz de indexar el texto completo de los campos que el administrador le especifique. Este proceso implica la creación de un fichero inverso, el cálculo de frecuencias o la eliminación de palabras vacías del texto. Todo ello se lleva a cabo de manera automática en el momento en que se almacena información en los campos definidos para este fin.

Particularidades del método FULLTEXT

Las opciones de búsqueda e indexación FULLTEXT, tienen múltiples características y aspectos que deben ser considerados. Por un lado efectúa un proceso automático de eliminación de palabras vacías y por otro lado consta de una serie de limitaciones que no aconsejan su uso para pequeñas colecciones de documentos.

- *Eliminación de palabras vacías*
  http://dev.mysql.com/doc/refman/5.0/en/fulltext-stopwords.html
  Cuando se efectúan búsquedas FULLTEXT, el fichero inverso de los textos indexados es tratado para eliminar las palabras vacías de la recuperación. También los términos de la consulta son tratados. Por defecto MySQL incluye tales palabras vacías en inglés en el archivo *ft_static.c*, por lo que la inclusión de un nuevo listado de palabras vacías conlleva su edición o la modificación de la ruta de la variable de configuración de MySQL *ft_stopword_file*.

- *Limitaciones*
  http://dev.mysql.com/doc/refman/5.0/en/fulltext-fine-tuning.html
  No todo son ventajas, ya que FULLTEXT también tiene limitaciones. En primer lugar la extensión de las palabras susceptibles de recuperación o consulta, tienen un límite de 3 caracteres, por lo tanto se deberán utilizar términos a partir de 4 o más caracteres. Esta propiedad puede ser editada desde la variable de configuración de MySQL *ft_min_word_len*. En segundo lugar las búsquedas en lenguaje natural se efectúan con un umbral de corte en torno al 50% de los términos indexados. Dicho de otro modo, elimina la mitad de los términos o

palabras a partir del análisis de frecuencias aplicando la técnica de cortes de Luhn, eliminando los términos más comunes y HAPAX.

Algoritmo de recuperación FULLTEXT en MySQL

http://forge.mysql.com/wiki/MySQL_Internals_Algorithms#Full-text_Search

http://www.miislita.com/term-vector/term-vector-1.html

MySQL utiliza un algoritmo de recuperación muy parecido al de ponderación de los términos mediante TF-IDF, es decir, frecuencia de aparición de los términos y frecuencia inversa del documento. TF-IDF es una medida de tipo estadístico utilizada para determinar la importancia de una palabra dentro de un documento en una colección o corpus documental. La importancia o peso del término se incrementa proporcionalmente al número de veces que una palabra aparece en el documento, compensándose con la frecuencia de la palabra en el corpus. Por ejemplo si un término aparece recurrentemente en el documento y a lo largo de toda la colección se obtiene una puntuación más baja. De hecho para evitar ese tipo de casos, MySQL procede a la eliminación de palabras vacías previamente al proceso de indexado.

Preparar la tabla en MySQL para FULLTEXT

La preparación de la tabla que almacena los registros y contenidos es clave para un buen funcionamiento del método FULLTEXT. Cuando se crea una tabla, el código para reseñar los campos que serán indexados a texto completo es FULLTEXT(campo1, campo2, campo3,...), véase tabla16.

```
CREATE TABLE comments (
id       INT NOT NULL AUTO_INCREMENT, PRIMARY KEY(id),
title    TEXT CHARACTER SET utf8 COLLATE utf8_general_ci,
user     VARCHAR(255) CHARACTER SET utf8 COLLATE utf8_general_ci,
date     VARCHAR(255) CHARACTER SET utf8 COLLATE utf8_general_ci,
comments  LONGTEXT CHARACTER SET utf8 COLLATE utf8_general_ci,
responses LONGTEXT CHARACTER SET utf8 COLLATE utf8_general_ci,
indexer   LONGTEXT CHARACTER SET utf8 COLLATE utf8_general_ci,
FULLTEXT(indexer)
) CHARACTER SET utf8 COLLATE utf8_general_ci
```

Tabla 16. Código para crear una tabla de comentarios en el que existe un campo indexer que almacenará todos los datos del comentario para su indexación

Aunque es posible utilizar múltiples campos para la recuperación a texto completo, se considera más eficiente crear un solo campo de tipo FULLTEXT expresamente

dedicado para la indexación de textos. Esto es lo que ocurre en el ejemplo anterior con el campo *indexer*, ya que almacenará el título de los comentarios, el nombre del usuario, los comentarios propiamente dichos y las respuestas a los mismos. También es muy recomendable tratar la información textual antes de que esta sea insertada en la tabla de la base de datos. Esto significa que siempre que sea posible se aconseja efectuar un proceso de eliminación de palabras vacías, sustitución de caracteres extraños por equivalentes en código HTML o ASCII, sustitución de caracteres acentuados por caracteres no acentuados, transliteración de caracteres, etc. Todo ello se puede conseguir utilizando junto con MySQL programas de tratamiento y depuración de textos, diseñados en muy diversos lenguajes de programación, por ejemplo PHP.

Búsquedas a texto completo con lenguaje natural

http://dev.mysql.com/doc/refman/5.0/en/fulltext-natural-language.html

Las búsquedas de tipo FULLTEXT se llevan a cabo utilizando dos cláusulas especiales. La cláusula MATCH() que indica entre paréntesis los campos indexados mediante FULLTEXT y la cláusula AGAINST() que contiene los términos de la consulta. Por defecto este tipo de consultas siempre son mediante lenguaje natural, esto es confrontar las palabras o términos de la consulta con la colección registrada en FULLTEXT. Este proceso se lleva a cabo por similaridad documental. El modelo de construcción de este tipo de consultas es el expresado en la tabla17.

```
SELECT * FROM catalogo WHERE MATCH(indexer) AGAINST('término/s de consulta')
```

Tabla 17. Consulta MATCH básica busca en lenguaje natural

Cuando se emplea la fórmula WHERE MATCH() los resultados devueltos se clasifican automáticamente por orden de relevancia. La relevancia se calcula en base al número de palabras del registro, el número de palabras únicas de ese registro, el número total de palabras en la colección, y el número de registros que contengan cada palabra determinada, esto es el modelo clásico de representación de documentos en el espacio vectorial.

Búsquedas booleanas a texto completo

http://dev.mysql.com/doc/refman/5.0/en/fulltext-boolean.html

Cuando se especifica en la cláusula AGAINST el atributo IN BOOLEAN MODE, se está indicando que la consulta en lenguaje natural adquiere propiedades booleanas. Esto habilita por ejemplo la posibilidad de decidir qué términos deben aparecer, cuáles no, o determinar qué frases deberán buscarse literalmente. Véase la sintaxis de la consulta en la tabla18.

```
SELECT * FROM catalogo WHERE MATCH(indexer) AGAINST('+t1 -t1 t3 >t4 <t5 (t6 t7 t8) ~t9 t10* "t11 t12 t13"' IN BOOLEAN MODE)
```

Tabla 18. Consulta FULLTEXT en modo booleano

El método IN BOOLEAN MODE, admite el empleo de modificadores para indicar operaciones muy precisas con los términos de consulta, a continuación se reseñan los más importantes:

- (+*término*) el signo más precediendo al término es el equivalente del operador AND e indica que obligatoriamente dicho término debe constar entre los resultados.

- (-*término*) el signo menos precediendo al término equivale al operador NOT e indica que el término no deberá figurar entre los resultados.

- (*término1 termino2*) si los términos no tienen ningún modificador o signo por defecto se emplea el operador OR, por lo que uno u otro término podrán figurar en los resultados.

- (>*término4*) un signo mayor que precediendo al término, le otorga un mayor peso en el cálculo de la relevancia, influyendo en un mayor número de resultados que contengan dicho término.

- (<*término5*) un signo menor que precediendo al término, le corresponde un menor peso en el cálculo de la relevancia, haciendo que los resultados tengan con menor frecuencia dicho término incluido.

- (*(término1 término2 término3)*) Cuando los términos están encerrados entre paréntesis, se indica a MySQL que deberán encontrarse lo más próximos posibles. Esto significa que los primeros resultados a mostrar serán aquellos que cumplan dicha condición.

- (*~término*) una tilde apaisada precediendo al término indica a MySQL que dicho término provoca ruido en la consulta, lo que le llevará a infraponderarlo para mejorar los resultados.

- (*término**) un asterisco ulterior al término se emplea a modo de truncamiento y concordará con aquellas palabras que empiecen por el término referido. Su empleo es de gran utilidad cuando se efectúan búsquedas a partir de las raíces de un término/s.

- (*"término1 término2 término3"*) Cuando varios términos están agrupados en un entrecomillado doble implica una búsqueda por frase exacta. Es muy importante reseñar el aspecto de la comilla doble en este caso en contraposición con la comilla simple que envuelve toda la consulta dentro de la cláusula AGAINST('consulta "subconsulta" ').

Búsquedas a texto completo con expansión de consulta

http://dev.mysql.com/doc/refman/5.0/en/fulltext-query-expansion.html

Las consultas basadas en FULLTEXT también soportan el método expansión de consulta, para recuperar la información. Esto es el empleo del algoritmo de retroalimentación automática por relevancia. Este funciona ejecutando dos consultas, por un lado la búsqueda con los términos de la consulta original y una segunda búsqueda en la concatena los términos de los documentos más representativos encontrados en la primera consulta. Para efectuar este tipo de consultas se añade el atributo WITH QUERY EXPANSION en la cláusula AGAINST(), véase tabla19.

```
SELECT * FROM catalogo WHERE MATCH(indexer) AGAINST('término/s de consulta' WITH QUERY EXPANSION)
```

Tabla 19. Consulta FULLTEXT con expansión de consulta

Búsquedas FULLTEXT con ranking de tipo SCORE

Para ordenar los resultados de las consultas en función del valor de relevancia obtenido en los procesos de recuperación, se necesita incorporar un campo temporal que almacene dicho valor. Esto se conoce como ranking de tipo score. Obsérvese la sintaxis de la consulta de la tabla20, se seleccionan los campos id, title, content y MATCH(campos) AGAINST('consulta') AS score. Esto significa que el coeficiente del ranking de la consulta expresada se almacenará en un campo que se ha resuelto llamar score (podría llamarse de cualquier otra forma, pero esta es la forma más común de denominarlo). A continuación el resto de la consulta es similar a las anteriormente expresadas, con la salvedad de que el ordenamiento puede realizarse por orden decreciente de relevancia e importancia desde el campo temporal score.

```
SELECT id, title, content, MATCH(indexer) AGAINST('término/s de consulta') AS score FROM catalogo WHERE MATCH(indexer) AGAINST('término/s de consulta') ORDER BY score DESC
```

Tabla 20. Consulta utilizando ordenación por ranking

6. Sistemas de clustering

Los sistemas de clustering son aquellos sistemas de recuperación que emplean algoritmos de agrupación de contenidos, por ello el proceso también puede adoptar otras denominaciones como categorización de los documentos de la colección.

> *"El concepto de clasificación de documentos refiere al problema de encontrar para cada documento la clase a la que pertenece, asumiendo que las clases están predefinidas y que se tienen documentos preclasificados para utilizar como ejemplos. En la presente tesis, se estudia la categorización o agrupamiento de documentos, entendiéndose por esto el proceso de encontrar grupos dentro de una colección de documentos basándose en las similitudes existentes entre ellos, sin un conocimiento a priori de sus características*." (GOLDENBERG, D. 2007)

Algunos de los algoritmos empleados para efectuar los procesos de agrupación son:

- *Categorización por objeto.* El objetivo es encontrar agrupaciones entre todos los documentos que conforman la colección. Esto significa que un porcentaje de términos relevantes de un grupo de documentos deberá estar presente en todos y cada uno de ellos.

- *Representación vectorial.* Cada documento de la colección se representa mediante vectores, quedando caracterizado por la frecuencia de aparición de los términos más relevantes y representativos. De esta forma se pueden comparar los vectores y agrupar los documentos en función de su similaridad.

    - *Cálculo del Centroide.* A partir de un grupo de documentos representados vectorialmente, se define el centroide que es el promedio de los vectores que componen el grupo.
    - *Reducción de términos a su raíz.*
    - *Eliminación de palabras vacías.*
    - *Eliminación de términos con bajo poder discriminatorio.*
    - *Eliminación de HAPAX.*

- *Similaridad documental.* Consiste en medir la distancia entre los vectores de cada documento para los que existen los algoritmos de:

    - *Coeficiente del coseno*. Cálculo del ángulo alfa. La semejanza entre los documentos se calcula como el producto vectorial entre ellos.
    - Otros: *Jaccard, Distancia Euclideana, Coeficiente de Dice, Sorensen, Hamming, Tversky*.

- *Métodos Jerárquicos.* Emplean algoritmos que permiten caracterizar los documentos de la colección con una estructura arbórea denominada dendograma, quedando definidos los grupos con cada vértice del árbol representado. A partir de la raíz del árbol (conformada por un único grupo que contiene todos los documentos), la división por grupos se produce cuando se analiza en el documentoA qué otro documento tiene mayor presencia sus términos.

- *Métodos Particionales.* En vez de trabajar a varios niveles para crear una estructura arbórea como en el caso anterior, se trabaja a un sólo nivel. Esto implica que el patrón de agrupación viene dado de antemano. Este factor establece las divisiones o partes con las que se calcula la similitud de los documentos.

- *Mapas auto-organizados.* También denominado sistema de redes neuronales.

Un ejemplo de Clustering: Carrot2

Carrot2 es un sistema de recuperación basado en técnicas de agrupación de documentos y contenidos web, sin requerir de bases de conocimiento externas como taxonomías o contenido preclasificado. Uno de sus algoritmos de agrupación es el correspondiente al método jerárquico, con los que es capaz de agrupar los contenidos de los motores de búsqueda Google o Bing. No obstante también puede emplearse para la recuperación de documentación dentro de un equipo cliente, siempre que disponga de una instalación previa "Google Desktop".

Para trabajar con Carrot2, se puede cargar su versión online desde el navegador web en la siguiente dirección: http://search.carrot2.org/stable/search. No obstante a efectos de probar todas sus posibilidades se recomienda la descarga de su versión carrot2-workbench-win32. Una vez descargado, descomprimir y ejecutar el archivo *carrot2-workbench.exe*.

- Opciones de búsqueda
    - *Fuente*. Se puede especificar qué base de conocimiento se desea utilizar para efectuar la búsqueda.
    - *Algoritmo*. Se permite la elección del algoritmo de agrupación de Carrot2. Por defecto se emplea Lingo, pero puede utilizarse K-mean, STC ó emplear los métodos habituales de agrupación por URL y fuente.
- Páginas de resultados
    - *Clusters*. Muestra un listado de todos los grupos identificados.
    - *Documentos*. Presenta un listado con los resultados más pertinentes de cada grupo.
- Visualización
    - *Esquema relacional Aduna*. (Aduna cluster map visualization). Muestra las relaciones entre unos grupos y otros.
    - *Diagrama circular de grupos*. (Circles visualization). Muestra una visión de los principales temas agrupados.
    - *Mapa de superficie por grupos*. (FoamTree visualization). Muestra un mapa de áreas con los grupos más pertinentes con colores cálidos en el

margen superior de la imagen y los grupos menos relevantes en el margen inferior destacados con colores fríos. En la base del dibujo aparece el grupo desconectado "Other Topics".

- Edición y configuración de atributos de consulta

  - *Grupos – Clusters*

    - *Cluster count base*: Número que establece el factor base para la creación de grupos. Cuanto mayor sea el número mayor será el número de grupos que generará. No existe equiparación entre este número y el número de grupos, significa que a partir del factor base se creará un número proporcional de grupos.

    - *Size-score sorting ratio*: Establece el equilibrio entre la puntuación de los grupos y el tamaño según cantidad de documentos. Si toma valor 0 el algoritmo ordenará según tamaño. Si toma el valor 1 los resultados se ordenarán en función de un ranking de puntuaciones. Un valor intermedio tendrá en cuenta ambos factores.

  - *Filtrado de etiquetas*. Efectúa un proceso de reducción de los términos, eliminación de palabras vacías, números, términos interrogativos, para efectuar posteriores procesos de indexación más eficaces.

  - *Etiquetas*

    - *Cluster label assignment method*

      - *Método único*. Asigna etiquetas únicas para cada vector en cada grupo o cluster. De esta forma evita duplicaciones de grupos. Por este motivo, al requerir contrastar todos los

vectores de todos los documentos entre sí, puede resultar un método lento pero más exhaustivo.

- *Método simple*. Asigna etiquetas en todos los vectores de cada grupo contrastándolos mediante similaridad documental, obteniendo grupos duplicados y no duplicados. En tal caso finalmente se eliminan aquellos grupos con etiquetas duplicadas, quedando un resultado más reducido. Se trata de un método rápido, pero menos exhaustivo.

- *Cluster merging threshold.* Es el porcentaje de coincidencia entre los documentos de dos cluster para que se fusionen en uno. Si se utilizan valores bajos, significa que los grupos tendrán un mayor nivel de coincidencia, con un corpus muy parecido. Cuando mayor es el valor, más riesgos de que el grupo sea más heterogeneo.

- *Phrase label boost*. Es el peso o puntuación específica que se otorga a varios términos cuando aparecen junto con otro. De esta forma se establecen relaciones de palabras o frases que siempre se recuperan juntas. Cuanto mayor sea el valor, mayor será la capacidad discriminatoria de las frases.

- *Phrase length penalty start*. Número de palabras máximo antes de ser infraponderada la frase o grupo de palabras.

- *Phrase length penalty stop*. Si la frase supera el número máximo de palabras será eliminada.

- *Title word boost.* Determina el peso que otorga a las palabras clave que coincidan con la consulta en el campo título.

- *Modelo de matriz*

  - *Factorization method*. Es el método de factorización de la matriz de documentos.

  - *Partial singular*. Factoriza en función del número máximo de vectores K.

  - *Factorization ED*. Tiene en cuenta todos los factores de configuración de etiquetas, filtrado y clusters.

  - *Factorization quality*. Es el número de iteraciones del proceso de factorización.

  - *Maximum matrix size*. Determina el número máximo de elementos de cada matriz de cada documento.

  - *Maximum word document frequency*. Determina la frecuencia máxima de las palabras en cada documento. Si la frecuencia supera a la especificada, la palabra será eliminada. El valor por defecto es 0,90 que indica 90%.

  - *Term weighting*. Determina el método para calcular el peso de las palabras del documento. Logaritmo de TF-IDF, Función lineal de TF-IDF o sólamente Factor TF.

- *Grupos multilingües*

  - *Default clustering language*. Idioma por defecto para efectuar el proceso de agrupación.

  - *Language aggregation strategy*. Define la estrategia de agregación idiomática. Se puede establecer que efectúe un

tratamiento de cluster para todos los idiomas, por idioma mayoritario ó creando grupos para cada idioma.

- *Extracción de frases*

  - *Phrase Document Frequency threshold*. Umbral de la frecuencia de aparición de frases en documentos. Las frases con una frecuencia de aparición menor a la indicada serán ignoradas.

  - *Truncated label threshold*. Umbral de truncamiento en etiquetas. Valores bajos determinan grupos más grandes, ya que el factor de truncamiento de términos es más alto.

- *Preprocesamiento*

  - *Exact phrase assignment*. Determina que los contenidos disponibles en cada agrupación coincidan con la consulta efectuada de forma exacta.

  - *Merge lexical resources*. Combina todas las palabras de todos los idiomas para formar parte del mismo lexicón de recuperación.

  - *Minimun cluster resources*. Define el número mínimo de documentos por grupo.

  - *Reload lexical resources*. Recarga todo el lexicón en cada consulta.

  - *Word Document Frequency threshold*. Umbral de frecuencia de los términos del documento, determina que cualquier término con un número de ocurrencias menor al especificado sea ignorado.

Referencias

- FIGUEROLA, C.G.; ALONSO BERROCAL, J.L.; ZAZO RODRÍGUEZ, A.F.; RODRÍGUEZ, E. 2002. Algunas Técnicas de Clasificación Automática de Documentos. Disponible en: http://multidoc.rediris.es/.../getdoc.php?id=90&article=28&mode=pdf

- GOLDENBERG, D. 2007. [Tesis Doctoral]. Categorización automática de documentos con mapas auto-organizados de Kohonen. Disponible en: http://www.itba.edu.ar/archivos/secciones/goldenberg-tesisdemagister.pdf

7. Sindicación de contenidos y recuperación de información

La sindicación de contenidos, también denominada redifusión de contenidos se emplea habitualmente en el marco de la transmisión de noticias para su lectura mediante diversos sistemas de lectura. La gran cantidad de información que se genera en estos medios hace necesario su conocimiento, de cara a su explotación documental, organización, clasificación y posterior recuperación. En el documento que se muestra a continuación se explican algunas de las bases sobre las que se alambica esta técnica.

Qué es sindicación de contenidos

Sindicación de contenidos es el proceso de redifusión de información que permite la suscripción a una fuente de información alimentada por sujetos productores de contenidos informativos, documentales en el corpus de un canal y un formato de datos que lo estructura para su intercambio, servicio, recopilación, lectura y gestión por parte de administradores, editores y usuarios.

- La sindicación es un proceso de comunicación que permite transmitir un contenido de un productor a múltiples usuarios.

- La sindicación es un conjunto de elementos que conforman una colección o canal de sindicación.

- La sindicación está basada en XML.

- La sindicación permite integrar contenidos en múltiples sitios web para su aprovechamiento.

- Los contenidos que se transmiten mediante sindicación son actualizados de forma constante o periódica.

- La naturaleza de los contenidos sindicados es variada, pudiendo ser cualquier tipo de dato o información.

- Existe una clara relación entre los sistemas de publicación y la sindicación como su herramienta de difusión.

- Existen herramientas que facilitan la lectura de la información transmitida mediante sindicación, haciendo alusión a lectores PARSER y agregadores.

También se entiende por sindicación, aquellas técnicas de redifusión de información estructurada mediante lenguajes extensibles de marcado, que logran configurar en su conjunto verdaderos canales o fuentes de información con entidad propia, de forma tal que posibilita la transmisión de sus contenidos a un usuario remoto mediante su suscripción, así como su consulta y actualización constante y periódica.

Aplicaciones habituales de la sindicación de contenidos

- Difusión de noticias y titulares mediante un canal de información de la unidad de información y documentación.
- Boletín de novedades bibliográficas.
- Difusión selectiva de categorías bibliográficas.
- Sindicación de resultados de búsquedas.
- Canales de sindicación de artículos de revistas.

Por qué es importante la sindicación de contenidos para la Documentación

La sindicación de contenidos permite transferir información de un sujeto productor a un usuario. Esto significa que cualquier biblioteca, archivo o centro de documentación, puede producir sus propios contenidos y hacerlos llegar a su público objetivo, a sus usuarios. Esta capacidad de hacer llegar la información supone en sí mismo el medio y canal más efectivo para estructurar y sistematizar la documentación de UID, para su redifusión.

Por otro lado, pueden crearse canales atendiendo a las múltiples necesidades documentales de una UID. De hecho pueden utilizarse los tradicionales formatos de sindicación como RSS1.0, RSS2.0 y Atom, para la representación de artículos, noticias, titulares u otros formatos como PRISM especializados en publicaciones seriadas,

MARC-XML para la sistematización de los catálogos bibliográficos u OPML para la agrupación de los canales de sindicación.

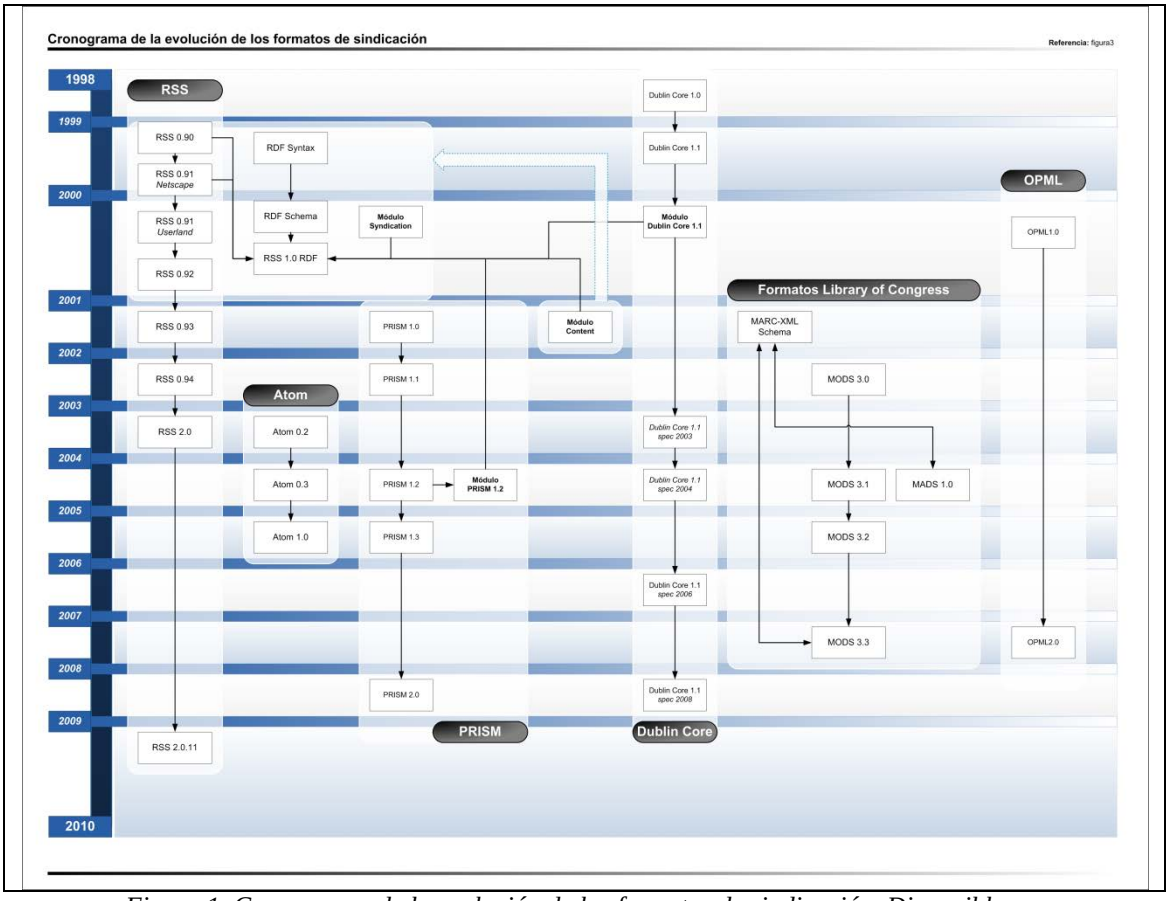

Figura 1. Cronograma de la evolución de los formatos de sindicación. Disponible en: http://www.mblazquez.es/blog-ccdoc-recuperacion/esquemas/esquema001.png

Dicho de otra forma, la sindicación posibilita describir, contener y referenciar cualquier tipo de documento que se encuentre en la biblioteca, siempre que el formato XML que se emplee en dicho cometido se adapte a sus principales rasgos descriptivos.

En qué se basa la sindicación de contenidos

La sindicación de contenidos, se basa en el empleo de las tecnologías de la web y más concretamente del lenguaje extensible de marcado XML. Es posible sindicar gracias al desarrollo de los formatos de sindicación, que no son más que un dominio de rasgos identificativos de un contenido o recurso, expresados a modo de etiqueta de marcado. Dicho de otra forma, los formatos de sindicación son fundamentalmente, lenguaje XML.

Cómo funciona la sindicación de contenidos

La sindicación de contenidos, como se ha explicado, aborda el proceso de difusión de unos contenidos generados por un sujeto productor. El sujeto productor puede variar según el dominio o entorno en el que se circunscriba. En el ámbito más general, se produce en los sistemas de publicación de la web, especialmente blogs, wikis, o CMS, que operan como plataformas de contenidos que son transmitidos libremente a través de la red. Dichos sistemas de publicación incorporan un programa denominado generador de canales de sindicación, con el que generan un archivo XML estructurado, de acuerdo con un determinado formato de sindicación, para permitir la redifusión y suscripción de los usuarios a dicho canal.

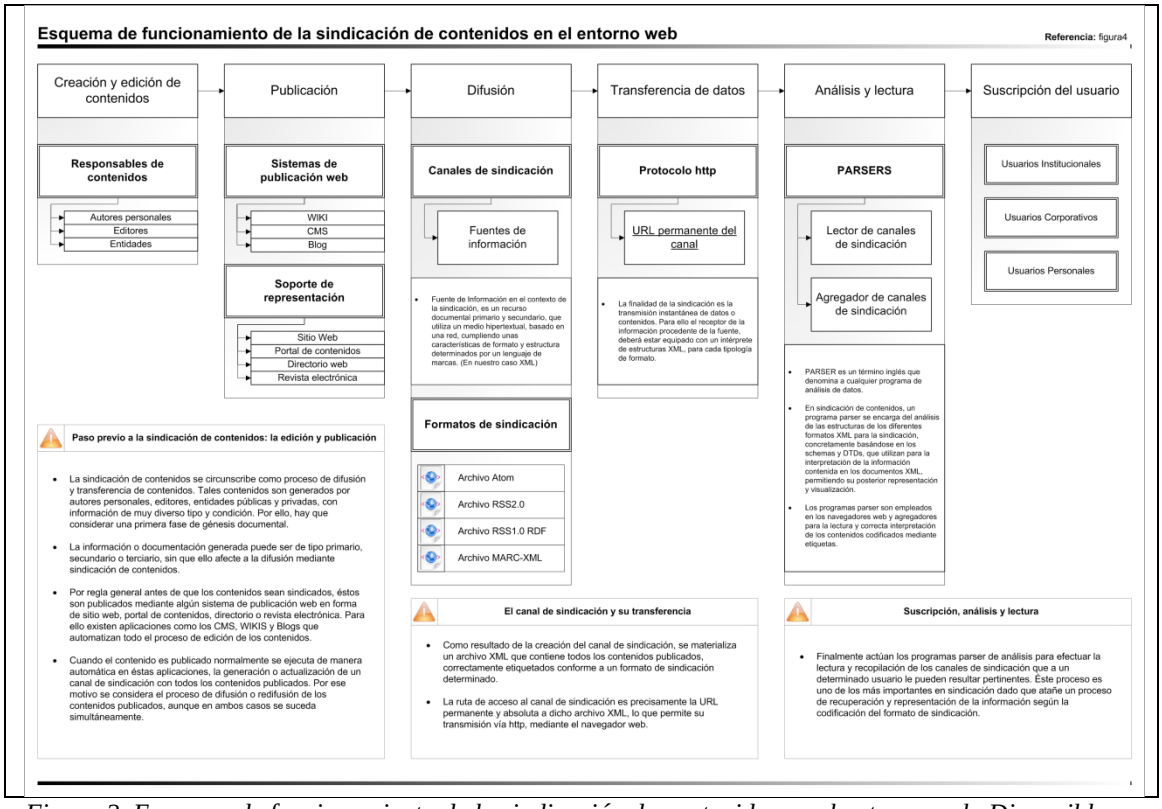

Figura 2. Esquema de funcionamiento de la sindicación de contenidos en el entorno web. Disponible en: http://www.mblazquez.es/blog-ccdoc-recuperacion/esquemas/esquema002.png

Tal proceso de redifusión es posibilitado en parte por los navegadores web que visualizan y representan los contenidos del canal de sindicación que contiene la información publicada en la web. Para que la información sea aprovechada y visualizada, existen programas denominados PARSER, que se encargan de analizar la estructura del archivo XML y extraer la información para recuperarla y representarla adecuadamente. Algunos ejemplos más comunes son los programas Google Reader,

FeedReader o FeedBurner, que además permiten recopilar los registros o elementos que componen cada canal al cual el usuario se suscribe. Cuando permiten esta opción de almacenamiento, también se los denomina programas agregadores.

Llegados a este punto del proceso en el que la información es transmitida al usuario, aún puede existir un mayor recorrido de la información contenida en el canal. De hecho el canal de sindicación puede ser reutilizado e incorporado en una tercera página web o bien mediante un programa PARSER, o bien mediante una hoja de estilo XSLT o CSS. Esta capacidad de transmitir un contenido, describirlo con un formato adecuado y aprovecharlo o bien mediante un lector, agregador, o bien mediante una tercera página web, ofrece una gran flexibilidad para transmitir información con fines documentales.

Fisionomía básica de un canal de sindicación

La fisionomía de un canal de sindicación hay que revisarla en su conjunto no sólo como un canal que constituye un único archivo, sino más bien como un conjunto de elementos que interactúan entre sí. De hecho la fisionomía de un canal de sindicación modelo es similar a la propuesta en el siguiente diagrama.

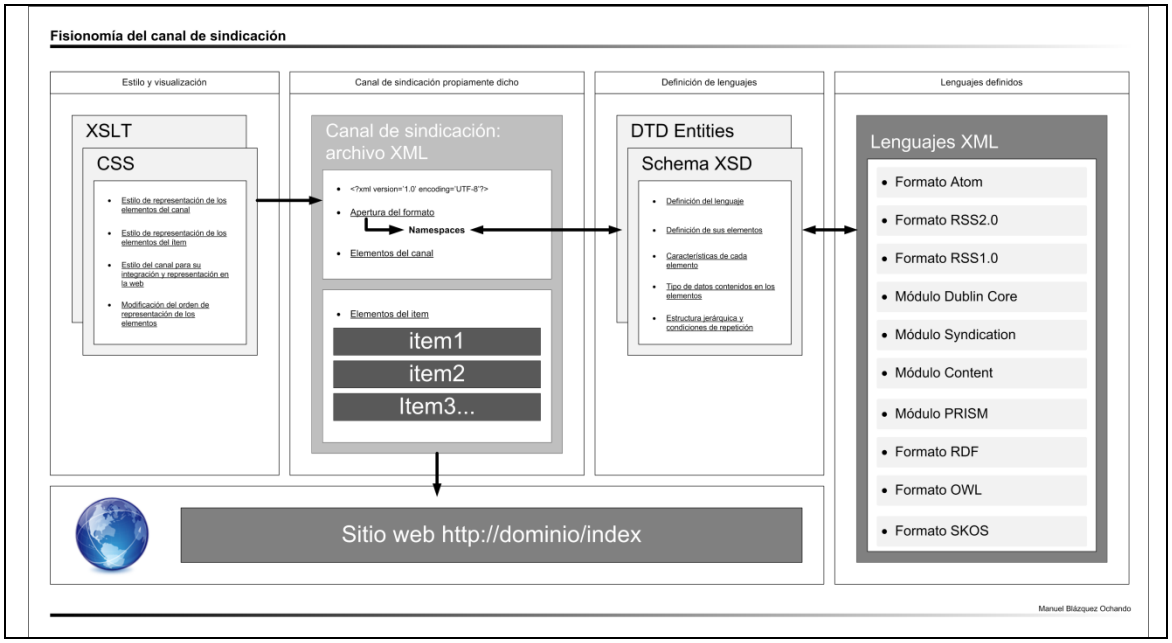

Figura 3. Fisionomía de un canal de sindicación. Disponible en: http://www.mblazquez.es/blog-ccdoc-recuperacion/esquemas/esquema003.png

Un canal de sindicación está compuesto por una cabecera de declaración de archivo XML *<?xml version='1.0' encoding='UTF-8'?>*. Posteriormente incluye la etiqueta

cabecera del formato de base con el que se constituye el canal como por ejemplo *<feed></feed>* en el caso del formato Atom, *<rss></rss>* en el caso del formato RSS2.0, *<rdf:RDF></rdf:RDF>* en el caso de RSS1.0 RDF. Tales cabeceras de formato permiten la introducción de atributos del tipo *xmlns=''* para declarar el espacio de nombres, también denominado Namespace de cada formato. Un ejemplo de namespace podría ser el que se muestra en la tabla21.

<rdf:RDF xmlns:rdf='http://www.w3.org/1999/02/22-rdf-syntax-ns#' xmlns='http://purl.org/rss/1.0/'>
Obsérvese en verde un Namespace adscrito al formato rdf y en amarillo un namespace propio del formato RSS1.0. En ambos casos son URLs absolutas

Tabla 21. Ejemplos de Namespace

El Namespace es la URI (Uniform Resource Identifier) de un archivo en formato XSD o DTD que contiene las especificaciones del lenguaje empleado en el formato de sindicación, de forma tal que se pueda validar su sintaxis. Dicho de otra forma, determinan como debe ser construido el formato atendiendo a la definición propia de cada elemento, el tipo de datos que deberá contener, los atributos aceptados para cada elemento, la repetición o no de determinadas etiquetas, su ubicación jerárquica en la estructura del formato. Por tanto un canal de sindicación mantiene relaciones con los archivos de definición de los lenguajes empleados.

Por otro lado el canal de sindicación puede enlazar archivos de estilo y visualización como CSS o XSLT, que permiten reproducir el contenido del canal de sindicación como si de una página web se tratase. Estos archivos inciden directamente en cómo se muestra la información en un navegador web.

Finalmente hay que considerar que el canal de sindicación es enlazado directamente a una página o sitio web del que depende directamente, constituyendo por si mismo la fuente de información de dicho recurso. El apartado de elementos de descripción del canal y de sus entradas o ítems de los que está formado o constituido.

Requisitos fundamentales de la sindicación de contenidos

- El formato de sindicación propiamente dicho deberá estar basado siempre en XML.
- El archivo XML debe estar bien formado y validado.
- La disposición de un canal o colección siempre contiene ítems o elementos jerárquicamente embebidos o anidados dentro del mismo.
- Los canales de sindicación deben enlazar esquemas de descripción de contenidos denominados Schemas XSD o DTD que permiten definir los elementos del formato empleado.
- Deberán disponer de hojas de formato y estilo adaptadas, desarrolladas en XSL preferiblemente, para su correspondiente visualización en cualquier navegador web.
- Permite la utilización del protocolo básico de comunicación web HTTP para la transmisión de datos mediante el método GET y POST o por medio del empleo de protocolos específicos de comunicación como SOAP o XML-RPC que permiten la recepción de una fuente de información determinada

Lo que nadie se atreve a decir sobre la sindicación y realmente se necesita conocer

Pese a todos los requisitos fundamentales de la sindicación de contenidos, la realidad es que no siempre los Namespaces de los formatos de sindicación se corresponden a un Schema XSD o DTD, tampoco suelen incluir archivos XSL para dar formato y estilo a los contenidos. Por si fuera poco, tampoco requieren necesariamente de protocolos SOAP o XML-RPC para mostrar los contenidos actualizados de un canal de sindicación, siendo únicamente necesario el protocolo HTTP. Incluso si el formato de sindicación no está validado pero sí bien formado, puede ser leído por la mayoría de los PARSER sin problemas. ¿Qué ocurre entonces con la sindicación de contenidos? ¿Qué diferencia habría entre un formato de sindicación y uno inventado por cualquiera? ¿Hasta qué punto la sindicación carece de normalización?

Ocurre que la sindicación de contenidos es un proceso completamente abierto a cualquiera que desee desarrollarlo con un nivel mayor o menor de exigencia en el cumplimiento de los requisitos fundamentales anteriormente mencionados. Pero además la implicación que ello tiene en los formatos de sindicación considerados por la

comunidad científica y la propia red, conlleva que cualquier formato que un usuario desarrolle con los requisitos fundamentales anteriormente mencionados, pueda utilizarse para sindicar contenidos, al cumplir la premisa de transmitir vía XML/HTTP un contenido estructurado, ser correctamente visualizado por un usuario remoto que disponga de un navegador web. En este sentido apenas existen diferencias. La diferencia habría que encontrarla en el uso que hacen los navegadores web de los formatos de sindicación, evidenciando que sólo unos pocos son los elegidos, concretamente Atom, RSS1.0 y RSS2.0, no sólo por sus consideraciones técnicas, sino más bien por la gran cantidad de usuarios que los utilizan. Por tanto se ha acuñado el concepto de sindicación reñido a unos pocos formatos con mayor proyección y utilización, generando una importante desigualdad y falta de correcto tratamiento para con los formatos menos conocidos que llevan a cabo tareas y operaciones similares. Dicho de otra forma, la sindicación de contenidos entendida como proceso de transmisión de contenidos, puede emplear cualquier formato XML, siempre y cuando éste pueda ser visualizado, representado y existan los mecanismos necesarios para generarlo y leerlo.

En cuanto al nivel de normalización de la sindicación, habría que distinguir el propio concepto de sindicación, las tecnologías de las que se nutre y el uso de las mismas. El concepto sindicación siempre ha partido del mismo punto y con el tiempo ha sufrido una evolución por adición de nuevos rasgos y características como la continua actualización de los contenidos, la reutilización del canal por terceros usuarios o la transmisión de los contenidos de un productor a un usuario destinatario. Todos estos elementos son partes integrantes de lo que puede hacer la sindicación, pero no es un concepto acotado, al ser en sí mismo un proceso comunicativo que se alimenta no sólo de la tecnología para llevar a efecto su objetivo, sino de los actores que vincula directa o indirectamente como los contenidos primarios, secundarios, sus productores, editores, destinatarios, usuarios, su reutilización, su retroalimentación, captura, lectura, modificación, conversión, exportación e importación.

En cuanto a la normalización tecnológica, hay que afirmar que existe y en gran medida, dado que XML constituye en sí mismo uno de los pilares de la web y está completamente definido. No es así si se desea normalizar los usos que se hacen de la tecnología, puesto que esta variable es libre y adopta formas completamente diferentes en función al objetivo a conseguir. De hecho si se limitase el modo de empleo de XML,

jamás evolucionaría la web de la forma en la que lo ha hecho y tampoco lo haría en el futuro. En consecuencia no se pueden normalizar los usos de una tecnología web como XML al igual que no se pueden normalizar los comportamientos del usuario ante un buscador y una demanda informativa. Por ese motivo si bien no se puede hablar de normalización de los usos que se hacen de la sindicación, si se puede tipificar en función a las experiencias que se obtienen en el desarrollo y evolución de la tecnología de sindicación ante una serie de objetivos concretos.

En conclusión, puede afirmarse que la sindicación de contenidos es un proceso abierto, normalizado en cuanto a su tecnología de base pero no en cuanto a su modo de empleo o procedimientos, lo que lo habilita para aprovecharse con fines biblioteconómicos y documentales.

Algunos conceptos útiles

- *Canal de sindicación*: Se denomina canal de sindicación a la fuente de información que se identifica como tal y que consta de ítems recogidos en un archivo XML estructurado conforme a un formato de redifusión de contenidos legible por un usuario, lector o PARSER. De esta forma es condición indispensable para un canal contener el título, denominación y descripción de la fuente de información que representa, el editor o editores responsables de los contenidos, su frecuencia de actualización o fecha de publicación y de forma obligatoria los contenidos debidamente estructurados y encapsulados.

- *Publicación del canal de sindicación*: Un canal de sindicación se considera a todos los efectos como tal, cuando está disponible su libre acceso para el usuario. Esto significa su representación y visualización, así como su posibilidad de suscripción. En este sentido un canal de sindicación podrá almacenarse en el gestor de marcadores del navegador por medio de suscripción o simplemente constar como un acceso directo al archivo XML contenedor de la fuente de información. En cualquiera de los casos resulta esencial la posibilidad de acceder a la información y representarla de forma básica y suficiente en sus etiquetas y elementos principales. También se considera esencial que junto con la publicación del canal esté disponga de un mecanismo de actualización

periódica constante, definido o parametrizado por el propio canal, siempre que el formato de redifusión así lo permita o por medio de la asistencia del propio sitio web.

- *Formato de sindicación*: El formato de sindicación es una estructura XML basada en un esquema XSD que posibilita la organización sistemática de la información y contenidos, así como su descripción e identificación. Dicho de otra forma los formatos aportan una interpretación de cómo ordenar y tabular las características básicas de un contenido. Por ejemplo, el título, el autor, la fecha de publicación, un resumen, los descriptores, el contenido propiamente dicho, contenidos relacionados, etc. Tales esquemas pueden adaptarse en mayor o en menor medida a las necesidades descriptivas o catalográficas de un tipo documental o informacional por lo que la elección de un formato u otro resulta trascendente para representar correctamente el contenido de la fuente de información. Actualmente existe conformidad en definir como formatos de sindicación de contenidos a RSS 1.0, RSS 2.0 y ATOM considerados además como los más comunes y populares de la web.

- *Item*: Se consideran ítems a cada unidad de contenido que conforma una fuente de información en un canal de sindicación. Tales unidades de información y contenido suelen estar estructuradas para definir y describir de forma pormenorizada sus elementos mínimos. Por ejemplo, si tomamos a una revista como una fuente de información con la que configurar un canal de sindicación, los ítems corresponderían a los artículos de los que está compuesta. A su vez cada artículo constaría de rasgos esenciales para su representación, como su título, subtítulo, mención de responsabilidad, fecha de publicación, resumen, palabras clave, contenido, páginas, volumen, número, ejemplar, sección, etc.

- *Parámetros de la sindicación*: Los parámetros de la sindicación son aquellos datos que configuran el funcionamiento o comportamiento de un canal de sindicación. Éstos parámetros son el periodo o tiempo de refresco para la actualización, el número de artículos o ítems visibles, la extensión del ítem (extendido, abreviado), el formato de sindicación y su esquema, la versión del formato de sindicación, los módulos y accesorios complementarios al esquema,

la interpretación de códigos HTML y la configuración de plantillas, PARSER o estilos de representación adjuntos.

Referencias: Atom

- The Atom Syndication Format. (2005). Network Working Group. The Internet Society. Disponible en: http://www.atomenabled.org/.../syndication/atom-format-spec.php

Referencias: RSS 0.X

- RSS 0.90 Specification. (1999) Netscape. Disponible en: http://www.rssboard.org/rss-0-9-0

- RSS 0.91 Specification. (2000). Userland Software. Disponible en: http://www.rssboard.org/rss-0-9-1

- LIBBY, D. (1999). RSS 0.91 Specification. Netscape. Disponible en: http://www.rssboard.org/rss-0-9-1-netscape

- RSS 0.92 Specification. (2000). Userland Software. Disponible en: http://www.rssboard.org/rss-0-9-2

Referencias: RSS 1.0 RDF

- GUHA. R. V.; BRICKLEY, D.; DAVIS, I. [et. al.] (2008). RDF Site Summary (RSS) 1.0. RSS-DEV Working Group. Disponible en: http://web.resource.org/rss/1.0/spec

- GUHA. R. V.; BRICKLEY, D.; DAVIS, I. [et. al.] (2001). RDF Site Summary 1.0 Modules. RSS-DEV Working Group. Disponible en: http://web.resource.org/rss/1.0/modules/

- GUHA. R. V.; BRICKLEY, D.; DAVIS, I. [et. al.] (2000). RDF Site Summary 1.0 Modules: Dublin Core. RSS-DEV Working Group. Disponible en:
- http://web.resource.org/rss/1.0/modules/dc/

- GUHA. R. V.; BRICKLEY, D.; DAVIS, I. [et. al.] (2000). RDF Site Summary 1.0 Modules: Syndication. RSS-DEV Working Group. Disponible en: http://web.resource.org/rss/1.0/modules/syndication/

- BEGED-DOV, B.; SWARTZ, A.; VLIST, E. (2002). RDF Site Summary 1.0 Modules: Content. RSS-DEV Working Group. Disponible en: http://web.resource.org/rss/1.0/modules/content/

- CROOME, C. (2002). RDF Site Summary 1.0 Modules: Qualified Dublin Core. Webarchitects. Disponible en: http://web.resource.org/rss/1.0/modules/dcterms/

Referencias: RSS 2.0.1

- WINER, D. (2003). RSS 2.0 at Harvard Law. Cambridge: Berkman Center for Internet & Society at Harvard University. Disponible en: http://cyber.law.harvard.edu/rss/rss.html

- WINER, D. (2003). RSS Advisory Board: RSS 2.0 Specification. Cambridge: Berkman Center for Internet & Society at Harvard University. Disponible en: http://www.rssboard.org/rss-2-0-1

Referencias: RSS 2.0.11

- WINER, D. (2009). RSS Advisory Board: RSS 2.0 Specification. Cambridge: Berkman Center for Internet & Society at Harvard University. Disponible en: http://www.rssboard.org/rss-specification

Referencias: OPML 1.0

- WINER, D. (2000). OPML 1.0 Specification. Disponible en: http://www.opml.org/spec

Referencias: OPML 2.0

- WINER, D. (2007). OPML 2.0 Specification. Disponible en: http://www.opml.org/spec2

8. Demostrador de procesos de sindicación de contenidos OrangeUp

La sindicación de contenidos es un proceso de comunicación y transmisión de dato ampliamente utilizado para efectuar el seguimiento de una serie de fuentes de información de forma sencilla y rápida. Dicho proceso es posible gracias a la disposición de una serie de programas capaces de generar los canales de sindicación en formato XML y a otros capaces de leer dichos canales en los formatos que se especifiquen. Dicho de otra forma, para que pueda darse una comunicación entre el emisor del canal de sindicación y el lector, ambos deberán compartir y entender el lenguaje en el que está codificada la información. En estos casos el lenguaje extensible de marcado XML ha dado lugar a terceros formatos RSS1.0, RSS2.0 y ATOM. Todos ellos son considerados por la comunidad científica como formatos de sindicación debido a que son legibles para la mayoría de los lectores de canales de sindicación, especialmente los que incorporan los navegadores web. Otra razón por la que se consideran formatos de sindicación es por su amplia difusión en el ámbito de los medios de comunicación y entre los usuarios que diariamente los utilizan.

La pregunta que hay que hacer llegados a este punto es: ¿Podría crearse un canal de sindicación con información bibliográfica? ¿Podrían utilizarse otros formatos derivados de XML, que al igual que RSS1.0, RSS2.0 y ATOM, permitieran la transmisión de registros bibliográficos, archivísticos o documentales? ¿Rigen las mismas normas de transmisión de datos para todos los formatos? ¿Las mismas técnicas de sindicación de contenidos pueden emplearse para otros casos? ¿Es cierto que con un generador de canales y un programa lector se pueden utilizar las mismas técnicas que se están utilizando para la sindicación de contenidos, empleando formatos especializados? Todas estas dudas y preguntas razonables, tienen su respuesta en el programa demostrador de procesos de sindicación de contenidos, OrangeUP, desarrollado exprofeso, para explicar el funcionamiento de las técnicas de sindicación, sus formatos, aplicaciones y realidades desconocidas para la comunidad científica.

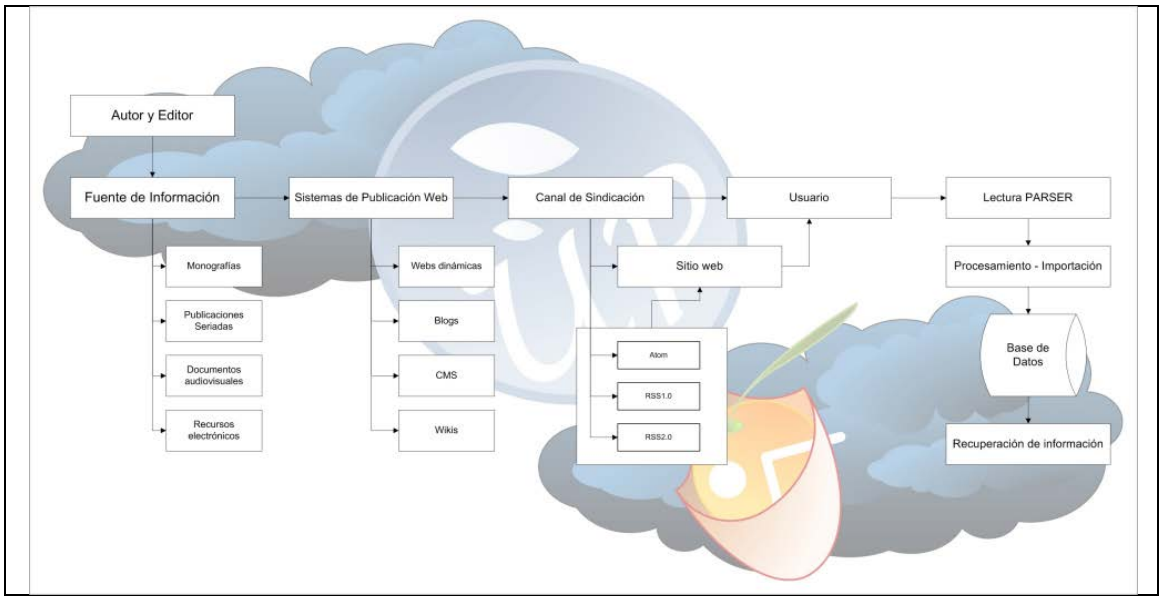

Figura 4. Proceso de publicación y difusión de canales de sindicación en RSS y MARC-XML

OrangeUP es un sistema de edición y gestión de canales de sindicación, que pretende abordar la experimentación de todo su ciclo. Para ello OrangeUP se ha diseñado para ser utilizado de forma aislada o en conjunto con terceros sistemas de publicación de manera que su integración sea lo más sencilla posible. El objetivo central de OrangeUP es servir de herramienta de aprendizaje, que facilite el mejor conocimiento del funcionamiento de los mecanismos de sindicación y redifusión de información, así como su aplicación a casos reales. OrangeUP también permite la elaboración de patrones de sindicación a medida, edición y publicación de contenidos, agrupación de canales de sindicación, lectura y selección de contenidos sindicados, así como su recuperación, véase figura4.

9. Sistemas de recuperación masiva basados en técnicas de sindicación de contenidos

Las técnicas de lectura y recuperación de canales de sindicación hacen posible el desarrollo de una nueva generación de buscadores especializados, muy parecidos conceptualmente a los tradicionales motores de búsqueda como Google, Yahoo, Bing y muy distintos en cuanto a su alimentación contextual y corpus documental. La principal diferencia reside en la selección de las fuentes de información, su control, descripción y recuperación de contenidos de forma exhaustiva y precisa. Ello hace posible que una búsqueda en MedWorm, sea más productiva para el colectivo de médicos y especialistas clínicos que por ejemplo en Google...

La sindicación de contenidos es clave para la las ciencias de la Documentación no sólo por el interés que suscita el mero hecho de poder controlar y desarrollar técnicas que permitan la transmisión de catálogos bibliográficos, registros, datos, o información. Es mucho más importante ser conscientes de que la mayor parte de los sitios web y sistemas de publicación digital tienen un canal de sindicación paralelo. Esto significa que una gran cantidad de información se está generando día a día, de forma limpia, resumida o completa, constantemente y cuyas fuentes de información pueden ser y son en muchos casos de gran importancia y relevancia. Millones de canales de sindicación y formatos que aún quedan por descubrir y desarrollar aguardan a que algún investigador o documentalista sea capaz de reconocerlos, emplearlos y aprovecharlos para hacer lo que siempre ha caracterizado a nuestra profesión, recuperarlos, documentarlos, describirlos y ponerlos al servicio de todos nuestros usuarios y lectores.

Finalmente, hay que recordar que todo buscador de esta naturaleza, así como de cualquier otra, emplea constantemente las técnicas de sindicación, agrupación (clustering), algoritmos de recuperación, SQL e indexación de los contenidos en centenares de bases de datos y clusters de almacenamiento repartidos en miles de servidores por todo el mundo. De tal forma que hasta este punto y según todo lo explicado hasta el momento, sea asienta la primera piedra de los conocimientos necesarios para comprender mejor y en algún momento configurar y desarrollar un verdadero sistema de recuperación de información.

10. Ejercicios prácticos

- Práctica1. Recuperación en MySQL
- Práctica2. Consultas Fulltext
- Práctica3. Asentando conocimientos de MySQL
- Práctica4. Recuperación con Carrot2
- Práctica5. Generación de canales de sindicación
- Práctica6. Lectura y recuperación de canales de sindicación

Práctica1. Recuperación en MySQL

Aprendida la teoría esencial para efectuar consultas de datos y contenidos en MySQL, se propone la resolución de una práctica en la que se pondrán en práctica todos los conocimientos aprendidos. Se deberá descargar un archivo SQL con estructura y datos correspondientes a un catálogo bibliográfico, instalar correctamente desde el gestor de bases de datos PhpMyAdmin, incluir un campo de identificación autonumérico para la identificación de los registros y finalmente responder a las preguntas y consultas que se plantean.

- Descargar archivo SQL de estructura y datos
- Descargar instrucciones de la práctica

1. Crea una base de datos denominada “prueba” o “test” e importa el archivo de datos *ucm1001ciencias.sql* en PhpMyAdmin.

2. Agregar campo “id” de tipo auto-numérico a la tabla importada.

3. Buscar todas las referencias de la editorial “elsevier” en cuyo título aparezca la palabra “curso”

Expresión SQL	
Nº de resultados	
Id de los registros	

4. Buscar todas las referencias cuyo título verse sobre “filosofía” y “ciencias sociales”

Expresión SQL	
Nº de resultados	
Id de los registros	

5. Buscar todas las referencias cuyo título verse sobre “ciencias sociales” pero no de “filosofía”

Expresión SQL	
Nº de resultados	
Id de los registros	

6. Buscar todos los registros con ISBN correspondiente a España.

Expresión SQL	
Nº de resultados	
Id de los registros	

7. Buscar todas las referencias de libros publicados por Edelvives o por el MEC, de forma absoluta.

Expresión SQL	
Nº de resultados	
Id de los registros	

8. Buscar todas las referencias de libros publicados por Edelvives o por el MEC, incluso si ambas participan en las mismas publicaciones.

Expresión SQL	
Nº de resultados	
Id de los registros	

9. Buscar todos los libros con más de 240 páginas.

Expresión SQL	

Nº de resultados	
Id de los registros	

10. Buscar todos los registros cuya fecha de publicación viene establecida por el Depósito Legal.

Expresión SQL	
Nº de resultados	
Id de los registros	

11. Buscar todas las sextas ediciones.

Expresión SQL	
Nº de resultados	
Id de los registros	

Práctica2. Consultas Fulltext

Conocida la sintaxis de consulta FULLTEXT, se propone la resolución de una práctica consistente en la experimentación con sentencias MATCH() AGAINST(). Se deberá descargar un archivo SQL con estructura y datos correspondientes a un servicio de información global, instalar correctamente desde el gestor de bases de datos PhpMyAdmin y finalmente responder a las preguntas y consultas que se plantean.

- Descargar archivo items1.sql
- Descargar archivo items2.sql
- Descargar archivo items3.sql
- Descargar instrucciones de la práctica2

1. Crea una base de datos denominada “prueba” o “test” e importa el archivo de datos *items1, items2 o items3 (El que se especifique en clase).*

2. Buscar “obama write in facebook”

Expresión SQL	
Nº de resultados	
Id de los registros	

3. Buscar “martin luther king and the civil rights”

Expresión SQL	
Nº de resultados	
Id de los registros	

4. Buscar “apple ceo founder steve jobs”

Expresión SQL	
Nº de resultados	
Id de los registros	

a. ¿Todos los resultados obtenidos corresponden a Steve Jobs?

Nº de resultados pertinentes	
Nº de resultados no pertinentes	

b. ¿Cómo reformularías la consulta en lenguaje natural para evitar el ruido en los resultados?

Expresión SQL	
Nº de resultados	
Id de los registros	

5. Buscar “arab spring in egypt”

Expresión SQL	
Nº de resultados	
Id de los registros	

a. ¿Cuántos resultados tienen que ver con la primavera árabe?

Nº de resultados sobre pertinentes	
Nº de resultados no pertinentes	

b. Reformular consulta con “conflict egypt” y razonar si los resultados obtenidos son más pertinentes. En segundo lugar razonar si el número de términos y su calidad afectan al resultado de las consultas

Nº de resultados sobre pertinentes	
Nº de resultados no pertinentes	
Razonar	

6. Buscar noticias de wall street, no relativas a las protestas y si es posible relacionadas con steve jobs y apple.

Expresión SQL	
Nº de resultados	
Id de los registros	

7. Buscar accidentes en las carreteras interestatales.

Expresión SQL	
Nº de resultados	
Id de los registros	

8. Buscar “hedge fund” por proximidad.

Expresión SQL	
Nº de resultados	
Id de los registros	

a. ¿Todos los resultados guardan el orden dado de las palabras?

Sí	
No	

b. En caso de añadirles el modificador + ¿Se recupera con el orden dado?

Sí	
No	

c. Reformular consulta para que los resultados guarden el orden exacto de las palabras dadas.

Expresión SQL	
Nº de resultados	
Id de los registros	

9. Buscar “germ” con truncamiento.

Expresión SQL	
Nº de resultados	
Id de los registros	

a. ¿Qué variaciones encuentras en los resultados obtenidos a partir de la raíz germ? Escribe los 5 primeros que encuentres.

1	
2	
3	
4	
5	

10. Buscar “Microsoft” con modo booleano.

Expresión SQL	
Nº de resultados	
Id de los registros	

a. Buscar “Microsoft” con expansión de consulta.

Expresión SQL	
Nº de resultados	
Id de los registros	

b. Qué método de búsqueda es más preciso en este caso.

booleano	
expansión	

11. Buscar “cars” con expansión de consulta.

Expresión SQL	
Nº de resultados	
Id de los registros	

a. A parte de resultados relativos a coches ¿se encuentran otros medios de transporte entre los resultados?

Sí	
No	

b. Reformular la consulta añadiendo un operador AND para limitar la búsqueda de expansión de consulta con otra de tipo booleano en la que se busque específicamente “trains”

Expresión SQL	
Nº de resultados	
Id de los registros	
¿Resulta útil la expansión de consulta para proponer resultados relacionados? Razona tu respuesta	

12. Retomar la consulta3 y añadir un ranking SCORE

Expresión SQL	
Nº de resultados	
Id de los registros	

Práctica3. Asentando conocimientos de MySQL

Con la intención de asentar los conocimientos adquiridos en cuanto a consulta de datos y recuperación de informaión se propone el desarrollo del siguiente ejercicio. Se deberá descargar el archivo SQL con estructura y datos correspondientes a un servicio de información global, instalar correctamente desde el gestor de bases de datos PhpMyAdmin y finalmente responder a las preguntas y consultas que se plantean.

- Descargar archivo items2000es.sql
- Descargar instrucciones de la práctica3

1. Crea una base de datos denominada “prueba” o “test” e importa el archivo de datos *items2000es*. Recuerda que pueden existir multitud de soluciones a los problemas planteados, procura resolverlos con lógica y aplicando los conocimientos aprendidos.

Búsquedas LIKE

2. Buscar noticias sobre la “OTAN”

Expresión SQL	
Nº de resultados	
Id de los registros	

a. ¿Todos los resultados obtenidos corresponden a la Organización del Tratado del Atlántico Norte?

Nº de resultados pertinentes	
Nº de resultados no pertinentes	

b. Si hay un resultado de un Porsche911, significa que algo ha fallado ¿pudo ser debido a LIKE? Recuerda cómo funciona LIKE y razona tu respuesta

3. Buscar todas las noticias sobre “tesoro” y “deuda”

Expresión SQL	
Nº de resultados	
Id de los registros	

4. Buscar todas las noticias sobre “España” desde el campo indexer.

Expresión SQL	
Nº de resultados	
Id de los registros	

a. ¿Aparece la palabra **España** o **Espana** entre los resultados obtenidos?

España	
Espana	

b. ¿A qué puede ser debido? Razona tu respuesta.

Búsquedas REGEXP

5. Buscar todas las noticias cuya fecha de publicación sea 21 de Octubre de 2011 ó el 22 de Octubre de 2011 ó el 23 de Octubre de 2001.

Expresión SQL	
Nº de resultados	
Id de los registros	

6. Buscar todas las noticias cuya enlace tenga presente el siguiente código “532830”.

Expresión SQL	
Nº de resultados	
Id de los registros	

Búsquedas en lenguaje natural

7. Buscar “rebaja de bonos de la eurozona”

Expresión SQL	
Nº de resultados	
Id de los registros	

8. Buscar “bonos de la eurozona” ó “agencias de calificación”

Expresión SQL	
Nº de resultados	
Id de los registros	

9. Buscar “energía nuclear”

Expresión SQL	
Nº de resultados	
Id de los registros	

a. ¿Todos los resultados obtenidos corresponden a energía nuclear?

Nº de resultados pertinentes	
Nº de resultados no pertinentes	

b. ¿Porqué hay un resultado sobre deportes?

c. ¿Cómo reformular la consulta para evitarlo?

Expresión SQL	
Nº de resultados	
Id de los registros	

Búsquedas en modo booleano

10. Buscar noticias sobre las elecciones del 20N y de dos de sus candidatos Rajoy y Rubalcaba, formular la consulta de forma tal que se recuperen más de 35 resultados.

Expresión SQL	
Nº de resultados	
Id de los registros	

11. Buscar noticias sobre la interpol y con especial valor de aquellas relativas a la justicia europea.

Expresión SQL	
Nº de resultados	
Id de los registros	

12. Buscar noticias relativas al Metro de Madrid.

Expresión SQL	
Nº de resultados	
Id de los registros	

13. Buscar noticias sobre “energía oscura”

Expresión SQL	
Nº de resultados	
Id de los registros	

Búsquedas con expansión de consulta

14. Buscar “videojuegos” con modo booleano.

Expresión SQL	
Nº de resultados	
Id de los registros	

a. Buscar “videojuegos” con expansión de consulta.

Expresión SQL	
Nº de resultados	
Id de los registros	

b. Qué método de búsqueda es más preciso en este caso.

booleano	
Expansión	

c. Combina ambas consultas para obtener resultados de la empresa de videojuegos “zynga” y “ea”.

Expresión SQL	
Nº de resultados	
Id de los registros	

Búsquedas con ranking

15. Retomar la consulta9 y añadir un ranking SCORE

Expresión SQL	
Nº de resultados	
Id de los registros	

Práctica4. Recuperación con Carrot2

Las técnicas de agrupación de contenidos pueden ser de gran utilidad para la recuperación masiva de documentos y su clasificación automática. Para poner a prueba los conceptos aprendidos, se propone la resolución de una práctica basada en un caso real "recuperación de empresas especializadas en ingeniería".

Objeto de búsqueda

1. Descargar el programa Carrot2 en su versión de aplicación de escritorio para Windows carrot2-workbench-win32. Una vez descargado, descomprimir y ejecutar. Siga las indicaciones que a continuación se reseñan para culminar la práctica.

2. El objeto de las consultas es la localización de empresas especializadas en "ingeniería". Las cadenas de consulta preferidas son las siguientes:
    a. Engineering companies
    b. Empresas de ingeniería
    c. Ranking de empresas de ingeniería
    d. Directorio de empresas de ingeniería
    e. Ranking of engineering
    f. Ranking of engineering enterprises
    g. Directory of engineering enterprises

Configuración básica

3. Configura el sistema con los siguientes parámetros:
    - Establecer la fuente de la búsqueda en "Bing"
    - Se utilizará el algoritmo de agrupación "Lingo" y "STC"
    - Aplicar valor market con el idioma correspondiente para cada cadena de consulta.
    - Número de resultados "10.000"

Ejecución de consultas

4. Ejecuta todas las cadenas de consulta propuestas para obtener el número de documentos y grupos con el algoritmo Lingo y STC. Reseña en la siguiente tabla los resultados.

Cadena de consulta	Lingo		STC	
	Nº de documentos	Nº de grupos	Nº de documentos	Nº de grupos
Engineering companies				
Empresas de ingeniería				
Ranking de empresas de ingeniería				
Directorio de empresas de ingeniería				
Ranking of engineering				
Ranking of engineering enterprises				
Directory of engineering enterprises				

5. Cuál de los dos algoritmos ofrece resultados más exhaustivos y cuál más pertinentes. Razona tu respuesta.

6. Qué cadena de consulta permite generar un grupo más pertinente de empresas especializadas en ingeniería. Copia los resultados

Cadena de consulta	
Grupo	
Resultados	[Pegar listado de resultados del grupo]

7. Se podrían obtener mejores resultados si se especificara el tipo de ingeniería. Razona tu respuesta.

Consulta y Refinamiento el sistema

8. Dada la generalidad de los resultados, se decide reconfigurar el sistema para mejorar su precisión. Llevar a cabo los siguientes cambios para efectuar consultas con el algoritmo "Lingo" sobre las cadenas de consulta propuestas.
   - Cluster count base: 40
   - Size-Score sorting: 0,9
   - Cluster merging threshold: 10
   - Phrase Label boost: 5
   - Title word boost: 10
   - Truncated label treshold: 0,63
   - Word document frequency threshold: 18

Cadena de consulta	Lingo	
	Nº de documentos	Nº de grupos
Engineering companies		
Empresas de ingeniería		
Ranking de empresas de ingeniería		
Directorio de empresas de ingeniería		
Ranking of engineering		
Ranking of engineering enterprises		
Directory of engineering enterprises		

9. ¿Se ha visto reducido el número de grupos y resultados? ¿Cómo han influido los factores que se han modificado en la configuración? Explica los cambios que han provocado en los resultados.

10. ¿Cuáles son los dos factores más determinantes de todos los que han sido editados y qué cambios se producen al editarlos?

Factores	Cambios que se producen

Práctica5. Generación de canales de sindicación

Para llevar a cabo la práctica se necesitará un usuario y contraseña para entrar en la aplicación OrangeUP, accesible desde http://www.mblazquez.es/testbench/orangeup. Resolver los siguientes apartados:

1. **Crear un canal de sindicación en formato RSS2.0**
   Hacer clic en la opción "Editar canales de sindicación >> Crear canal RSS2.0"

2. **Crear un canal bibliográfico en formato MARC-XML**
   Hacer clic en la opción "Editar canales de sindicación >> Crear catálogo MARC"

3. **Añadir 3 noticias al canal RSS2.0**
   Hacer clic en la opción "Editar registros de canales >> Crear item"

4. **Añadir 3 registros bibliográficos al canal MARC-XML**
   Hacer clic en la opción "Editar registros de canales >> Crear record"

NOTA: Para todos los casos se deberá rellenar todos los campos de descripción, siempre que sea posible, se recomienda utilizar informaciones de casos reales (canal de noticias de un medio de comunicación, registros de un catálogo bibliográfico real). Por otro lado se recomienda evitar el uso de caracteres extraños (#, @, €, &, ', ") debido a que el programa OrangeUP aún está en fase de desarrollo y ello podría causar algún problema en la resolución de los ejercicios de la práctica

Práctica6. Lectura y recuperación de canales

El segundo elemento esencial para demostrar un proceso de sindicación de contenidos es la disposición de un lector de canales de sindicación, capaz de interpretar el lenguaje de cada formato RSS1.0, RSS2.0 y ATOM. Con la práctica6 se evidenciará irrefutablemente que MARC-XML no sólamente puede ser generado y compartido como cualquier canal de sindicación, sino que también puede ser capturado, recuperado y leído perfectamente con un programa parser similar al que se utiliza para todos los demás formatos de sindicación. De esta forma se demuestra que la técnica de sindicación de contenidos puede ser empleada también para otras actividades y finalidades como las bibliográficas, archivísticas, biblioteconómicas o documentales.

- Descargar programas parser de prueba
- Descargar instrucciones de la práctica6

Para llevar a cabo la práctica se necesitará un usuario y contraseña para entrar en la aplicación OrangeUP, accesible desde http://www.mblazquez.es/testbench/orangeup. A continuación descargar desde el blog el archivo "parsers-mblazquez.zip" que contienen los programas que se utilizarán para efectuar las pruebas con canales de sindicación desde la aplicación OrangeUP.

1. **Captación de los datos de un canal de sindicación con el método DOM**

- Hacer clic en la opción "Banco de pruebas" y crear una nueva prueba con el código fuente del archivo "prueba-mapa-dom.php"
- Probar el programa **prueba-mapa-dom** e imprimir pantalla con el resultado obtenido.

Probar e imprimir pantalla	
[pegar aquí]	
¿Qué formato se está visualizando?	

- Cambiar el valor de la variable $feed (Revisar el código fuente que se ha cargado en el archivo de prueba) por el siguiente: http://mblazquez.es/docs/marc.xml

Probar e imprimir pantalla
[pegar aquí]

2. Lectura de formatos ATOM

- Hacer clic en la opción “Banco de pruebas” y crear una nueva prueba con el código fuente del archivo “prueba-parser-atom.php”
- Probar el programa **prueba-parser-atom** e imprimir pantalla con el resultado obtenido.

Probar e imprimir pantalla
[pegar aquí]

- Cambiar el valor de $feed por un canal de sindicación de Google News en formato ATOM

URL del canal de Google News
[pegar aquí]
Probar e imprimir pantalla
[pegar aquí]

3. Lectura de formatos RSS2.0

- Hacer clic en la opción “Banco de pruebas” y crear una nueva prueba con el código fuente del archivo “prueba-parser-rss2.php”
- Probar el programa **prueba-parser-rss2** e imprimir pantalla con el resultado obtenido.

Probar e imprimir pantalla
[pegar aquí]

- Cambiar el valor de $feed por la URL del canal de sindicación RSS2.0 que se creó en la práctica anterior.

URL del canal RSS2.0 creado en la práctica5
[pegar aquí]
Probar e imprimir pantalla
[pegar aquí]

4. Lectura de formatos MARC-XML

- Hacer clic en la opción "Banco de pruebas" y crear una nueva prueba con el código fuente del archivo "prueba-parser-marc.php"
- Probar el programa **prueba-parser-marc** e imprimir pantalla con el resultado obtenido.

Probar e imprimir pantalla
[pegar aquí]

- Cambiar el valor de $feed por la URL del canal MARC-XML que se creó en la práctica anterior.

URL del canal MARC-XML creado en la práctica5
[pegar aquí]
Probar e imprimir pantalla
[pegar aquí]

11. Índice de tablas

12. Índice de figuras

13. Bibliografía y referencias

- BEGED-DOV, B.; SWARTZ, A.; VLIST, E. 2002. RDF Site Summary 1.0 Modules: Content. RSS-DEV Working Group. Disponible en: http://web.resource.org/rss/1.0/modules/content/
- BLÁZQUEZ OCHANDO, M. 2010. [Tesis Doctoral]. Aplicaciones de la sindicación para la gestión de catálogos bibliográficos. Disponible en: http://eprints.ucm.es/11233/1/T32065.pdf
- BLÁZQUEZ OCHANDO, M. 2010. [Software online]. OrangeUP: demostrador de procesos de sindicación de contenidos. Disponible en: http://mblazquez.es/testbench/orangeup/
- CROOME, C. 2002. RDF Site Summary 1.0 Modules: Qualified Dublin Core. Webarchitects. Disponible en: http://web.resource.org/rss/1.0/modules/dcterms/
- CUERDA GARCÍA, X.; MINGUILLÓN ALFONSO, J. Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto. Mosaic, 2004, nº 36.
- DOLAN, F. 2011. [Software online]. Medworm. Disponible en: http://www.medworm.com/
- FIGUEROLA, C.G.; ALONSO BERROCAL, J.L.; ZAZO RODRÍGUEZ, A.F.; RODRÍGUEZ, E. 2002. Algunas Técnicas de Clasificación Automática de Documentos. Disponible en: http://multidoc.rediris.es/.../...?id=90&...=28&=pdf
- FRANGANILLO, J.; CATALAN, M.A. Bitácoras y sindicación de contenidos: dos herramientas para difundir la información. BiD, 2005, diciembre.
- FRIEDL, J. 2006. Mastering Regular Expressions: Understand Your Data and Be More Productive. Disponible en: http://www.minek.com/files/Mastering%20Regular%20Expressions.pdf
- GOLDENBERG, D. 2007. [Tesis Doctoral]. Categorización automática de documentos con mapas auto-organizados de Kohonen. Disponible en: http://www.itba.edu.ar/archivos/secciones/goldenberg-tesisdemagister.pdf
- GUHA. R. V.; BRICKLEY, D.; DAVIS, I. [et. al.] 2000. RDF Site Summary 1.0 Modules: Dublin Core. RSS-DEV Working Group. Disponible en:

- GUHA. R. V.; BRICKLEY, D.; DAVIS, I. [et. al.] 2000. RDF Site Summary 1.0 Modules: Syndication. RSS-DEV Working Group. Disponible en: http://web.resource.org/rss/1.0/modules/syndication/
- GUHA. R. V.; BRICKLEY, D.; DAVIS, I. [et. al.] 2001. RDF Site Summary 1.0 Modules. RSS-DEV Working Group. Disponible en: http://web.resource.org/rss/1.0/modules/
- GUHA. R. V.; BRICKLEY, D.; DAVIS, I. [et. al.] 2008. RDF Site Summary (RSS) 1.0. RSS-DEV Working Group. Disponible en: http://web.resource.org/rss/1.0/spec
- http://web.resource.org/rss/1.0/modules/dc/
- LIBBY, D. 1999. RSS 0.91 Specification. Netscape. Disponible en: http://www.rssboard.org/rss-0-9-1-netscape
- Resource Description Framework (RDF). Disponible en http://www.w3.org/RDF/
- RIBEIRO-NETO, B.; BAEZA-YATES, R. Modern Information Retrieval. Addison-Wesley, 1999.
- RSS 0.90 Specification. 1999 Netscape. Disponible en: http://www.rssboard.org/rss-0-9-0
- RSS 0.91 Specification. 2000. Userland Software. Disponible en: http://www.rssboard.org/rss-0-9-1
- RSS 0.92 Specification. 2000. Userland Software. Disponible en: http://www.rssboard.org/rss-0-9-2
- SKINNER, G. 2011. RegExr. Disponible en: http://gskinner.com/RegExr/
- The Atom Syndication Format. 2005. Network Working Group. The Internet Society. Disponible en: http://www.atomenabled.org/.../syndication/atom-format-spec.php
- WINER, D. 2000. OPML 1.0 Specification. Disponible en: http://www.opml.org/spec
- WINER, D. 2003. RSS 2.0 at Harvard Law. Cambridge: Berkman Center for Internet & Society at Harvard University. Disponible en: http://cyber.law.harvard.edu/rss/rss.html

- WINER, D. 2003. RSS Advisory Board: RSS 2.0 Specification. Cambridge: Berkman Center for Internet & Society at Harvard University. Disponible en: http://www.rssboard.org/rss-2-0-1
- WINER, D. 2007. OPML 2.0 Specification. Disponible en: http://www.opml.org/spec2
- WINER, D. 2009. RSS Advisory Board: RSS 2.0 Specification. Cambridge: Berkman Center for Internet & Society at Harvard University. Disponible en: http://www.rssboard.org/rss-specification